UNITED STATES PATENT APPLICATION

A METHOD OF ASSEMBLING AUTHORIZATION CERTIFICATE CHAINS

INVENTOR

Victor B. Lortz

Schwegman, Lundberg, Woessner & Kluth, P.A.
1600 TCF Tower
121 South Eighth Street
Minneapolis, MN 55402
ATTORNEY DOCKET SLWK 884.501US1
Client Ref. No. P11597

A METHOD OF ASSEMBLING AUTHORIZATION CERTIFICATE CHAINS

Background

[0001] Computer security provides confidentiality, authentication, integrity, and non-repudiation. Confidentiality prevents an eavesdropper from reading a message. Authentication prevents an intruder from masquerading as the originator of a message. Integrity prevents an intruder from modifying a message in transit or substituting a false message for a legitimate one. Non-repudiation prevents a sender from falsely denying later that he sent a message.

[0002] Encrypting a message is one way to ensure confidentiality. Figure 1 is a block diagram of a typical encryption method in the prior art. Suppose Alice wants to send a confidential message to Bob. First, Bob sends Alice his public key 102. Then, Alice creates a cleartext 104 (an unencrypted message) encrypts it using Bob's public key 102, and sends the resulting ciphertext 106 to Bob. Next, Bob decrypts the ciphertext 108 with his private key 110, enabling him to read the cleartext 112. Eve, an eavesdropper, is unable to read the ciphertext 106 during transit and cannot decrypt it with the public key, so it is confidential. Suppose, however, when Bob initially sent his public key 102 to Alice, Eve intercepted Bob's public key 102 and substituted her own public key. Alice—thinking she had Bob's public key—encrypted the message and sent it to Bob. Eve intercepted again and decrypted the ciphertext 106 and read the cleartext 112. Even worse, Eve then substituted different cleartext using Bob's public key and Bob thought it came from Alice. How can Bob tell if the message came from Alice? Bob can tell by authenticating with an identity certificate.

[0003] Figure 2A is a block diagram of an identity certificate 202 in the prior art. The identity certificate 202 associates an identity 204 with a public key 206 to ensure the public key 206 belongs to the identity 204. Identity certificates 202 contain information from a Certification Authority (CA) 208 and a digital signature 210. The digital signature 210 is proof of authorship by the person identified in the identity 204. If Bob digitally signs 210 an identity certificate 202 through a CA, who verifies his identity, Eve is thwarted from substituting her public key 102 for

Bob's. This is because Alice verifies Bob's public key 102 using his identity certificate 202, before encrypting the cleartext 104. Thus, the identity certificate 202 provides authentication. It also ensures integrity and non-repudiation.

[0004] However, identity certificates 202 have significant drawbacks in certain practical situations. Suppose E-Commerce Inc. forms a business relationship with Major Corporation. Later, E-Commerce outsources one aspect of its business to Fly-By-Night Consulting Ltd. Major issues identity certificates to E-Commerce providing E-Commerce secure access to protected resources, including a customer database. In turn, E-Commerce issues identity certificates to Fly-By-Night and gives Fly-By-Night a copy of its identity certificate from Major. In order for Fly-By-Night to use the identity certificates issued by E-Commerce to access Major's customer database, Fly-By-Night needs a copy of the identity certificate issued from Major to E-Commerce to complete the identity certificate chain. In short, Fly-By-Night must present both its identity certificate and E-Commerce's identity certificate to Major to access the customer database. Major decides whether or not to allow access to Fly-By-Night based on additional information, such as a mapping of certificate identities to permissions. But, this is a burden for Major to maintain. Thus, Major is tempted to take shortcuts like accepting all certificates signed by E-Commerce. There is a need for a new way to assemble certificate chains so that E-Commerce retains more control over Fly-By-Night's access to Major's resources.

[0005] Sharing full security information with a subcontractor is an unacceptable risk, because there is no long-term trust relationship. E-Commerce needs a more secure method for Fly-By-Night to communicate its privileges to Major, one that preserves the trust relationship with Major. Trust is essential to doing business, especially over the Internet.

[0006] On the Internet, distributed services traditionally have been based on Remote Procedure Call (RPC), such as Distributed Component Object Model (DCOM) and Common Object Request Broker Architecture (CORBA). These used network security mechanisms that are not valid across domains, such as Windows NT and login domain. Also, firewalls routinely block RPC connections, because of the security threat they represent. A firewall is a system that prevents access to or from a private computer network. Web services attempted to finesse these problems with

Hypertext Transfer Protocol (HTTP) connections not blocked by firewalls. However, this attempt did not solve the underlying issue of establishing cross-domain trust and authorization.

[0007] On the contrary, web services potentially open up huge security risks. These risks are not adequately addressed by current security mechanisms. For example, these risks are not adequately addressed by web services defined as Extensible Markup Language (XML) based Simple Object Access Protocol (SOAP) interfaces. SOAP interfaces are associated with Universal Resource Identifiers (URIs) that are accessible via Hyper Text Transfer Protocol (HTTP). XML defines standard formats for sharing information. To add authentication to SOAP packets, SOAP communications can be signed using XML digital signatures. The XML digital signature specification defines a standard way to transmit digital certificates for both conventional X.509 identity certificates and for new Simple Public Key Infrastructure (SPKI) authorization certificates. X.509 and SPKI are standards for defining digital certificates. Unfortunately, little attention has been given to the new SPKI authorization certificates. New methods should take advantage of the new SPKI authorization certificates, because the conventional X.509 identity certificates have significant practical drawbacks. Web services need new security mechanisms that directly address the underlying issue of establishing cross-domain trust and authorization. For example, E-Commerce needs to establish cross-domain trust and authorization with both Fly-By-Night and Major.

[0008] Instead of turning over its identity certificate to Fly-By-Night, E-Commerce should have delegated restricted or limited privileges to Fly-By-Night in order to protect Major's protected resources. Identity certificates 202 do not permit delegation, restrictions, or limitations. Also, E-Commerce shouldn't have to tell Major about the outsourcing. Subcontractors frequently change. Tracking and updating the identity certificates of subcontractors creates a nuisance of overhead expenses for Major. Yet, identity certificates 202 require the identity 204 of subcontractors like Fly-By-Night to be known, tracked, and updated by Major. E-Commerce needs a better method of authorizing a certificate for Fly-By-Night, one that allows E-Commerce more control over communication between Fly-By-Night and Major. E-Commerce and other organizations need certificates capable of

delegating limited privileges to a third party, without revealing the identity 204 of the third party, while still providing confidentiality, authentication, integrity, and non-repudiation.

Brief Description of the Drawings

[0009] Figure 1 is a block diagram of a typical encryption method in the prior art.

Figure 2A is a block diagram of an identity certificate in the prior art.

Figure 2B is a block diagram of an authorization certificate to be contrasted with the identity certificate shown in Figure 2A.

Figure 3 is a block diagram of an authorization certificate chain.

Figures 4A and 4B are block diagrams that together show a method embodiment of the present invention for assembling authorization certificate chains among an authorizer, a client, and a third party.

Figure 5 is a block diagram of a data signal embodiment of the present invention that is sent from a client to a third party.

Figure 6 is a block diagram of an example embodiment of the data signal shown in Figure 5 as a Simple Object Access Protocol (SOAP) request and an example method of using it.

Figures 7 and 8 are flow charts of additional method embodiments of the present invention for assembling authorization certificate chains among an authorizer, a client, and a third party.

Detailed Description

[0010] Methods for assembling authorization certificate chains and associated data signals are described. In the following detailed description, reference is made to the accompanying drawings, which form a part hereof. These drawings show, by way of illustration, specific embodiments in which the invention may be practiced. In the drawings, like numerals describe substantially similar components throughout the several views. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. Other embodiments may be used and structural, logical, and electrical changes may be made without departing from the scope of the present invention.

[0011] Figure 2B is a block diagram of an authorization certificate 220 to be contrasted with the identity certificate 204 shown in Figure 2A. The identity certificate shown in Figure 2A associates an identity 204 with a public key 206, while the authorization certificate 220 shown in Figure 2B associates an authorization 222 with a public key 224. Thus, an authorization certificate 220 issued to a third party does not reveal the identity of the third party. The authorization certificate 220 also contains delegated privileges 226 and a validity interval 228. Delegated privileges 226 may include conditions, limitations, and restrictions on those privileges, including scope, time, and context. The validity interval 228 defines if and when the authorization certificate is valid.

[0012] Traditional identity certificates 204 have significant drawbacks, while authorization certificates 220 are capable of delegating limited privileges to a third party, without revealing the identity 204 of the third party, while still providing confidentiality, authentication, integrity, and non-repudiation. Traditional identity certificates 204 are implemented according to the X.509 standard, while authorization certificates 220 are implemented according to the SPKI standard. SPKI certificates can directly encode authorizations, while X.509 certificates merely bind public keys to identities. Software applications must subsequently interpret X.509 identities to make authorization decisions using mechanisms subject to additional security risk, such as compromising a database mapping certificate identities to login account names. SPKI certificates also support constrained delegation of authority.

[0013] The delegation features of SPKI certificates enable certificate issuers to specify whether the entity to which an authorization certificate is issued is capable of empowering other keys to obtain the same or more limited authorization. These features enable web service providers to use subcontractors to perform specific aspects of the service on behalf of the client without the client needing to know of or issue credentials to the subcontractors. Interestingly, many of the scenarios used to describe and motivate the use of web services describe the same "service is a client of other services" model. However, the trust and security issues that arise in this context are an open issue. For example, suppose a client authorizes a service to access one of its own protected resources, e.g. a client-side database by agreeing on

a shared secret or digital certificate value. Current web service proposals have no good way for the service to delegate access to subcontractors. Divulging the shared secret or other credential to a subcontractor is unacceptable, because there is often no long-term trust relationship with the subcontractor. SPKI certificates provide an elegant and robust solution to this open issue.

[0014] Returning to the Fly-By-Night scenario in the background, suppose E-Commerce issued an authorization certificate 220 to Fly-By-Night instead of an identity certificate 202. Fly-By-Night no longer has direct access to the certificate issued from Major to E-Commerce and, thus, cannot carry out a bypass attack. In addition, E-Commerce has increased control over Fly-By-Night's use of the delegated privileges 226 through conditions, limitations, and restrictions and the validity interval 228.

[0015] Figure 3 is a block diagram of an authorization certificate chain 300. An authorizer is an entity that makes an authorization decision based on a key used to sign a certificate that is recognized, known, and trusted by the authorizer. Any holder of an SPKI cert6ificate can act as an authorizer. An authorization certificate chain 300 is a sequence of one or more certificates issued by a holder of authorized keys. The chain conveys the authorization where the root key is trusted by prior knowledge. For example, Major has a root key stored on a secure computer system used in creating certificates. In the Fly-By-Night scenario, Major is the authorizer and signer of an authorizer-to-client certificate 304 and E-Commerce signs the client-to-third-party certificate 306.

[0016] Figure 3 shows a certificate chain 300 from an authorizer 302 to an authorizer-to-client certificate 304 to a client-to-third party certificate 306. In the Fly-By-Night scenario, the authorizer 302 is Major, the authorizer-to-client certificate 304 is the certificate issued from Major to E-Commerce, and the client-to-third party certificate 306 is the certificate issued from E-Commerce to Fly-By-Night. When Fly-By-Night exercises the privileges in the client-to-third party certificate 306, E-Commerce is notified to control the access and to provide the authorizer-to-client certificate 304. Thus, Fly-By-Night never has knowledge of the authorizer-to-client certificate 304, keeping E-Commerce in the loop. In addition, E-Commerce is in the loop so that it can reconsider the access at the time it is being

exercised and, if necessary, revoke the privilege. For example, E-Commerce might revoke a privilege if Fly-By-Night was no longer doing business with E-Commerce or if other circumstances had changed.

[0017] For example, SOAP requests for web services are augmented with XML signatures containing SPKI certificates in the SOAP headers. These certificates authenticate and authorize the SOAP request to perform some operation on the service. So far, this is just the standard usage model for signed XML with SPKI certificates. However, in some cases multiple SPKI certificates may be needed to complete the authorization. This situation arises whenever SPKI delegation is used. With SPKI delegation, the authorizer signs a delegation-enabled certificate empowering a client to perform some operation and the client signs another certificate empowering a third party to perform the same or a more limited operation. For the third party to demonstrate its right to perform the operation, both certificates must be submitted to the authorizer. In the traditional model, the third party will have both certificates in its possession and will include both of them in its SOAP request. The present invention describes an alternative model where the original SOAP request does not contain all the needed certificates.

[0018] In the present invention, the third party's certificate by itself is insufficient to obtain the privileges granted by the client. However, when the two certificates, the certificate issued from the authorizer to the client and the certificate issued from the client to third party, are combined, the third party is granted the privileges. In the present invention, the certificates are combined in a new way. Conventionally, the authorizer maintains copies of the certificates issued to the client. But, this is undesirable for two reasons. First, it creates additional maintenance overhead for the authorizer. Second, it prevents the client from imposing additional constraints on the third party's use of the delegation authorization.

[0019] Unlike X.509 identity certificates, SPKI authorization certificates do not need CAs and do not require Certificate Revocation Lists (CRLs). A CRL is a mechanism that a CA uses to publish and disseminate information about revoked certificates to certificate-issuers. A CRL is analogous to a list of bad checking accounts maintained by a grocery store. Maintaining and publishing notices of revoked certificates is widely acknowledged to be expensive and problematic. One

aspect of the present invention is a method where an intermediary, e.g. a client, issues a long-lived SKPI authorization certificate to a third party so that the validity of the certificate and corresponding potential for harm is implicitly limited by the subsequent involvement of the intermediary in the certificate processing. This increases security in cases where services are outsourced across organizational boundaries where trust relationships are less stable than in traditional business models.

[0020] Figures 4A and 4B are block diagrams that together show a method embodiment of the present invention for assembling authorization certificate chains among an authorizer 400, a client 402, and a third party 404. Figure 4A shows how the certificates are initially setup. First, an authorizer 400 delivers 406 a first certificate to a client 402. For example, Major delivers a first certificate to E-Commerce. Then, the client 402 stores 408 the first certificate and a Universal Resource Indicator (URI) that is associated with a third party 404 in a database 410. Next, the client 402 delivers 412 a second certificate and the URI to the third party 404.

[0021] Figure 4B shows what happens when the third party 404 requests access to a protected resource using the second certificate. The third party 404 passes 414 the second certificate and URI to the authorizer 400. Next, the authorizer 400 performs an HTTP "Get" 416 on the URI. Then, the client 402 retrieves 418 the first certificate using the URI and performs additional processing. As a result, the client returns 420 the first certificate to the authorizer 400. Finally, the authorizer 400 grants the request 422.

[0022] Figure 5 is a block diagram of a data signal embodiment of the present invention that is sent from a client to a third party. One aspect of the present invention is a data signal 500 sent from a client to a third party. The data signal comprises a second digital certificate issued from the client to the third party 502 and a URI 503. The URI is capable of retrieving a first digital certificate from a database associated with the client, wherein the first digital certificate issued from an authorizer to the client. Figure 5 shows an example where the URI 503 uses the Internet 506 to execute a client script 508 to retrieve an authorizer-to-client certificate 510. The client script is an Active Server Page (ASP) or a Common

Gateway Interface (CGI) program or some other code executable over the Internet. In one embodiment, the second digital certificate grants less power than the first digital certificate. In another embodiment, the first and second digital certificates are SPKI certificates. SPKI certificates are examples of the type of certificate capable of performing the method of the present invention and other kinds of digital certificates having cryptographic techniques capable of expressing the methods of the present invention also work.

[0023] Figure 6 is a block diagram of an example embodiment of the data signal shown in Figure 5 as a Simple Object Access Protocol (SOAP) request 600 and an example method of using it. The SOAP Request 600 comprises header data 602 that comprises an SPKI certificate 604 and a Retrieval Method including a URI 606. First, the third party 404 signs 608 the SOAP request and sends it to the authorizer 400. The authorizer 400 extracts 610 the SPKI certificate from the SOAP request and performs an HTTP "Get" 612 on the URI. The URI invokes 614 a client web server 616 that performs additional processing 618 and returns 620 a certificate issued by the authorizer to the client 622.

[0024] Figures 7 and 8 are flow charts of additional method embodiments 700, 800 of the present invention for assembling authorization certificate chains among an authorizer, a client, and a third party. As shown in Figure 7, one aspect of the present invention is a method for assembling authorization certificate chains among an authorizer, a client, and a third party 700. First, the client stores at least one first certificate from the authorizer 702. Also, the client stores a URI associated with both the at least one first certificate and the third party 704. Then, the client provides at least one second certificate and the URI to the third party 706. When the authorizer accesses the URI 708, the client provides the at least one first certificate to the authorizer 710, so that the client retains control over the third party's use of the first certificate 712.

[0025] In one embodiment, the client provides a third certificate with a short-term usage to the third party, upon demand by the authorizer. In another embodiment, the third certificate is a one-time use certificate. In another embodiment, the third certificate is a short-lived certificate. In another embodiment, the one-time use or short-lived certificate is generated on demand. In another embodiment, the one-time

use or short-lived certificate is generated on demand by the authorizer. For example, suppose Major issued a long-lived certificate to E-Commerce and E-Commerce issued a long-lived certificate to Fly-By-Night, but E-Commerce also issued a short-lived or one-time use certificate to Fly-By-Night. This forces Fly-By-Night to go back to E-Commerce after a short time or one use to get another certificate, giving E-Commerce more control. The short-lived or one-time use certificate could be generated during the additional processing of step 6 of Figure 4B and returned to the authorizer along with the first certificate in step 7. Short-lived and one-time use certificates prevent bypassing of the client by the third party. In another embodiment, a secure channel is established with the authorizer in place of a one-time use certificate.

[0026] In another embodiment, when the authorizer accesses the URI, the client authenticates the authorizer. In another embodiment, the client limits the third party's use of the first certificate. In another embodiment, the client tracks the third party's use of the first certificate. In another embodiment, the contents of the first certificate are not revealed to the third party. In another embodiment, when the authorizer accesses the URI, the client does not provide the certificate when it is requested, effectively revoking the privilege. In another embodiment, the third party delegates part or all of its privileges to a fourth-party, so that the third party would be acting as the client in the methods of the present invention with respect to the fourth party.

[0027] As shown in Figure 8, another aspect of the present invention is a machine-accessible medium having associated content capable of directing the machine to perform a method of assembling authorization certificate chains among an authorizer, a client, and a third party 800. In one embodiment, the associated content is a software development kit. The client receives a first certificate from the authorizer 802 and then generates a URI associated with both the at least one first certificate and the third party 804. The client provides a second certificate and the URI to the third party 806. After the third party provides the second certificate and URI to the authorizer 808, the authorizer access the URI 810 and, then, the client provides the first certificate to the authorizer 812. In one embodiment, the third party provides the second certificate and URI to the authorizer in an XML signature.

In another embodiment, the first and second certificates are SPKI certificates. In another embodiment, the authorizer grants access to the third party. In another embodiment, the client tracks at least one use of the second certificate. In another embodiment the client revokes the second certificate.

[0028] In an example embodiment of the present invention, the client maintains a database of third parties to which it has issued authorization certificates. The database contains enough information to construct certificates and store them. Other information, such as the client's arrangement with each third party and information related to later processing is optionally included in the database. Each such certificate is identified by a certificate identification value (certID). The certID is scoped by the particular context of the client and third party, so it need not be globally unique. When the client issues a certificate to the third party, it also provides the third party with a URI string of the form URI_prefix#certID. For example, the third party may provide the URI string http://www.TheClient.com/SubDomain/SPKI/Auth1234#delegate5678. The URI can also contain information about the third party and other information to help the client web server later on when it processes the request for that URI. When the third party signs a SOAP request to send to the authorizer, it first includes its own SPKI certificate. For example, the third party includes its own SPKI certificate in an <SPKIData> element of the XML signature's <KeyInfo> element. To complete the authorization information for the signature, the third party also includes a <RetrievalMethod> element of the signature's <KeyInfo> with the URI value specified by the client. For example, <RetrievalMethod URI=http://www.TheClient.com/SubDomain/SPKI/Auth1234delegate5678/>.

[0029] When the authorizer receives the SOAP request, it extracts the SPKI certificate and also performs an HTTP Get operation on the RetrievalMethod's URI according to the XML signature specification. This operation invokes a web server belonging to the client and supplied the context in which to process the request. The client optionally performs any additional processing steps and then returns the SPKI certificate(s) issued to it by the authorizer. For example, the client performs additional processing to verify the status of its relationship with the third party, logs the date and time of the request, maintains a count of the number of such requests

and other similar acts.

[0030] It is to be understood that the above description it is intended to be illustrative, and not restrictive. Many other embodiments are possible and some will be apparent to those skilled in the art, upon reviewing the above description. For example various types of digital certificates may be used in place of SPKI certificates, languages other than XML may be used, communications other than SOAP requests may be used, Universal Resource Locators (URLs) may be used in place of URIs, Intranets, Local Area Networks (LANs) or other networks may be used in place of the Internet, and more. Therefore, the spirit and scope of the appended claims should not be limited to the above description. The scope of the invention should be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.